Projet commun MIF12-MIF16

Erwan Guillou, Jean-Michel Moreau

18 septembre 2007

Un projet combiné entre les modules ‘Compilation’ et ‘Gestion de Projets’ est demandé aux étudiants de M1. Il consiste en la réalisation d’un travail en groupes de 4 à 5 personnes sur environ 3 mois (octobre à décembre).

Table des matières

1 Sujets

Le sujet est à choisir parmi les suivants :

1. Écriture d’un générateur d’analyseur lexical. Ce générateur fonctionnera sur les mêmes principes que `flex` : même format de fichier d’entrée, même protocole de communication avec `bison`, il devra de plus permettre d’exporter une description de l’automate fini déterministe. Il est aussi demandé de développer une interface graphique permettant de visualiser l’automate et le fonctionnement de l’analyseur lorsque l’utilisateur entre des informations au clavier.
2. Écriture d’un générateur d’analyseur syntaxique (*cf.* `bison`). Le format des fichiers d’entrée sera le même que celui de `bison`. Dans un premier temps, les étudiants s’interesseront à la générations d’analyseurs à partir de grammaires LL(k) sous la forme d’un analyseur procédural. Puis à la génération d’analyseur plus ’haut’ niveau, à partir de grammaires LR sous la forme d’un automate à pile.
3. Réalisation d’un « shell » (Linux) : commandes de base + interprétation de scripts shell + utilisation des variables d’environnement. Plus de détail en cours MIF12. Dans un premier temps, vous mettrez en oeuvre les commandes de base disponibles dans n’importe quel shell : ls, cd, pwd, echo, help, history, alias, unalias et exit. Dans un second temps, on souhaite être capable de gérer les variables d’environnement par le biais des commandes : set, unset et env. Fichier de script + fichier de config (*cf.* `.bashrc`). Références : page man de bash, GNU bash manual
4. Analyse de flot de données : à partir d’un sous-ensemble du langage C++, il est demandé de générer les graphes d’inclusion, d’appel de fonctions ainsi que le graphe de flot de donnée d’une fonction ou d’un programme. Au cours du projet, le sous-ensemble du C++ étudié sera étendue de manière à prendre en compte, au final, la majeure partie du langage C++.
5. Interpréteur `Lisp`. Cet interpréteur, écrit en C++, implémentera la syntaxe de base de Common Lisp (*cf.* Lessons 1-7, 11 du livre sur Common Lisp[1]).

[1] dept-info.labri.fr/~strandh/Teaching/Programmation-Symbolique/Common/David-Lamkins/cover.html

6. Conversion de formats de fichiers XML → LaTeX. Ce projet consiste à proposer un formalisme XML pour la représentation de documents. Dans un premier temps, l'application développée devra convertir des fichiers issus de ce formalisme en une représentation LaTeX du document. Dans un second temps, vous devrez développer une application permettant de faire la conversion inverse (passer de LaTeX vers XML).
7. Outil de présentation fondé sur un formalisme de description à définir dans le cahier des charges ; génération directe de PostScript à partir du formalisme, et retour de PostScript (issu du générateur) vers le formalisme initial. Pour les plus motivés : on pourra généraliser en extrayant la structure de tout fichier postscript vers le formalisme choisi et défini dans le projet.
8. Pour les étudiants ne faisant que MIF16 : écriture d'un tableur (*cf.* gnumeric) limité à certaines fonctionnalités (cahier des charges à négocier en MIF16). Si des étudiants en situation symétrique prennent le sujet suivant, la collaboration active des deux groupes est indispensable.
9. Pour les étudiants ne faisant que MIF12 : implémenter le langage de macros pour le tableur réalisé ci-dessus (cahier des charges à négocier en MIF12). Travail en collaboration avec le groupe précédent (sous la supervision des deux chefs de projet).
10. Les deux sujets ci-dessus pourront aussi être réalisés par un seul groupe d'étudiants devant passer MIF12 et MIF16.
11. Possibilité de proposer un sujet, qui devra être visé par les deux enseignants pour être validé. Le sujet doit obligatoirement revêtir un aspect compilation.

2 Modalités

- Chaque projet (sauf projets 8 et 9) doit impliquer de 4 (minimum) à 6 étudiants (maximum).
- Un étudiant ne peut participer qu'à un seul projet.
- Chaque projet aura un coordinateur parmi les étudiants, et deux chefs de projet : Erwan Guillou et Jean-Michel Moreau.
- Chaque projet sera réalisé en heures hors emploi du temps. Il s'accompagnera d'un rapport. Le code, en C/C++, sera documenté (`Doxygen`) et s'accompagnera d'une documentation externe (manuel de l'utilisateur), les deux rédigées en LaTeX et générée en `pdf`.
- La partie « compilation » des projets pourra être travaillée sur certains créneaux de TP de MIF12. La partie « gestion de projets » sera traitée en réunions hors emploi du temps, à décider avec l'enseignant.
- Chaque projet sera géré par `cvs`, en utilisant l'espace commun de stockage maintenu par les administrateurs pour le M1 (précisions en cours MIF16). Un suivi régulier des archives cvs sera fait par les chefs de projet.
- Le projet (sauf projets 8 et 9) est noté dans les deux disciplines, au titre du contrôle continu. Il est évalué sur le rendu (cahier des charges, manuel d'utilisateur, programme exécutable, sources documentées, rapport en LaTeX, exposé de présentation unique par groupe et devant les enseignants des deux disciplines, etc), et démonstration. Cette note de projet est combinée aux éventuelles notes de CC données en TD ou en TP dans les deux matières.

3 Calendrier approximatif (à venir)

- Détermination des listes de sujets / groupes : semaine prochaine en TD/TP.
- ébauche de cahier des charges : à renvoyer par le coordinateur du groupe vers mi-octobre. Cette ébauche a pour but de nous permettre de recentrer le travail quand nécessaire.
- version finale du cahier des charges vers fin octobre.
- programmation etc : novembre.
- rendu des projets (archives, documentations, rapports...) : avant la fin des enseignements du semestre (mi-décembre).

– présentations des projets et sessions de démonstration : juste avant vacances de Noël.

Des réunions intermédiares auront lieu entre responsables de groupes et les deux chefs de projet, afin d'assurer le suivi et l'évolution des projets (dates à fixer ultérieurement).

4 Recommandations

- C'est une bonne idée de mettre en place un wiki (dokuwiki est très simple à installer, si besoin demander de l'aide) sur l'un de leur page web (avec suivi du projet dessus, ...). La première des choses à mettre dessus serait un lien vers le document que vous êtes en train de lire. Pensez aux aspects collaboratifs : qui aura le droit de modifier, etc.
- L'exposé TD sur SPIP (espace collaboratif) pourrait aussi servir de point de départ à une réflexion sur le travail collaboratif dans ce projet. Rappel : il faut 3 exposés sur SPIP (20 mn max., un par groupe, 3 étudiants par exposé) pour le premier TD de Gestion de Projets, dans 15 jours. Concertez-vous et régissez vite, mais pas dans le désordre : organisez-vous, et qu'une seule personne de la promo m'envoie par mél les noms des candidats retenus pour cet exposé.. Rappelez-vous : les sujets de mél sont très importants (genre [M1-GP] noms pour pour exposé SPIP du 27/09).
- Une autre bonne idée est de mettre en place un serveur d'archivage (svn, cvs) sur un de vos comptes. Donc, le deuxième sujet d'exposé pour le même jour (et 9 autres candidats à trouver) est un « must » : même procédure que ci-dessus.